

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting a data object having an identifier (ID) from the first storage location; storing the ID in a second lock object;

determining whether the ID is stored successfully in the second lock object and, upon a successful storage, storing the ID in a first lock object, thereby indicating that the data object is stored at the first storage location;

storing the data object, ~~the ID of which is contained in the first lock object~~, at the second storage location;

deleting the data object, ~~the ID of which is contained in the first lock object~~, from the first storage location;

deleting the ID from the first lock object, thereby indicating that the data object is not stored at the first storage location, after the ~~respective data object assigned to that ID~~ has been deleted from the first storage location; and

deleting the ID from the second lock object after the ID has been stored in the first lock object.

2. (Previously Presented) The method of claim 1, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.
3. (Currently Amended) The method of claim 1, wherein the data object is stored in a file and wherein an assignment of the ID to the file or to a name of the file, ~~in which the data object assigned to the ID is stored~~, is stored in the first lock object.
4. (Previously Presented) The method of claim 1, wherein the first lock object is stored on a nonvolatile storage means.
5. (Currently Amended) The method of claim 1, wherein the ID is stored in the second lock object after selecting [[a]] the data object having an ID from the first storage location ~~for the respective data object~~.
6. (Currently Amended) The method of claim 1, wherein the ID ~~of the selected data object~~ is stored in the second lock object before the data object ~~assigned to that ID~~ is stored at the second storage location.
7. (Currently Amended) The method of claim 1, wherein storing the ID in the first lock object further comprises:
~~storing IDs of all selected other data objects in the first lock object before storing any of the selected data object objects at the second storage location.~~

8. (Currently Amended) The method of claim 1, further comprising:
checking whether the ID for the data object has been stored in the first lock
object, and if the ID has been stored, skipping storing the data object at the second
storage location.

9. (Previously Presented) The method of claim 1, further comprising:
checking whether the data object is contained in the second storage location,
and if the data object is contained in the second storage location, skipping storing the
data object at the second storage location.

10. (Previously Presented) The method of claim 9, wherein the checking is
performed by querying the first lock object.

11. (Currently Amended) The method of claim 1, further comprising:
~~determining whether the data object was stored in the first lock object~~
~~successfully, and upon unsuccessful storage, checking whether the data object~~
~~assigned to the respective ID has been stored in the second storage location, and if the~~
~~data object respective ID has not been stored, skipping deleting the data object from the~~
~~first storage location and skipping deleting the ID from the first lock object.~~

12. (Previously Presented) The method of claim 1 for use in an enterprise
resource planning software.

13. (Currently Amended) A computer system for processing data, the computer system comprising:

memory means for storing program instructions;

input means for entering the data;

storage means for storing the data;

a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for moving data objects in the computer system from a first storage location to a second storage location, the method comprising:

selecting a data object having an identifier (ID) from the first storage location;

storing the ID in a second lock object;

determining whether the ID is stored successfully, and upon a successful storage, storing the ID in a first lock object, thereby indicating that the data object is stored at the first storage location;

storing the data object, ~~the ID of which is contained in the first lock object~~, at the second storage location;

deleting the data object, ~~the ID of which is contained in the first lock object~~, from the first storage location;

deleting the ID from the first lock object, thereby indicating that the data object is not stored at the first storage location, after the respective data object assigned to that ID has been deleted from the first storage location; and

deleting the ID from the second lock object after the ID has been stored in the first lock object.

14. (Canceled).

15. (Currently Amended) A computer readable storage medium comprising instructions for performing a method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting a data object having an identifier (ID) from the first storage location;

storing the ID in a second lock object;

determining whether the ID is stored successfully, and upon a successful storage, storing the ID in a first lock object, thereby indicating that the data object is stored at the first storage location;

storing the data object, ~~the ID of which is contained in the first lock object~~, at the second storage location;

deleting the data object, ~~the ID of which is contained in the first lock object~~, from the first storage location;

deleting the ID from the first lock object, thereby indicating that the data object is not stored at the first storage location, after the respective data object assigned to that ID has been deleted from the first storage location; and

deleting the ID from the second lock object after the ID has been stored in the first lock object.

16. (Canceled).

17. (Currently Amended) The computer readable storage medium of claim 15, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

18. (Currently Amended) The computer readable storage medium of claim 15, wherein the data object is stored in a file and wherein an assignment of the ID to the file or to a name of the file, ~~in which the data object assigned to the ID is stored~~, is stored in the first lock object.

19. (Currently Amended) The computer readable storage medium of claim 15, wherein the first lock object is stored on a nonvolatile storage means.

20. (Currently Amended) The computer readable storage medium of claim 15, wherein the ID is stored in the second lock object after selecting [[a]] the data object ~~having an ID from the first storage location for the respective data object~~.

21. (Currently Amended) The computer readable storage medium of claim 15, wherein the ID ~~of the selected data object~~ is stored in the second lock object before the data object ~~assigned to that ID~~ is stored at the second storage location.

22. (Currently Amended) The computer readable storage medium of claim 15, wherein storing the ID in the first lock object further comprises:

storing IDs of ~~all selected other~~ data objects in the first lock object before storing ~~any of the selected data object objects~~ at the second storage location.

23. (Currently Amended) The computer readable storage medium of claim 15, wherein the method further comprises:

checking whether the ID ~~for the data object~~ has been stored in the first lock object, and if the ID has been stored, skipping storing the data object at the second storage location.

24. (Currently Amended) The computer readable storage medium of claim 15, wherein the method further comprises:

checking whether the data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping storing the data object at the second storage location.

25. (Currently Amended) The computer readable storage medium of claim 24, wherein the checking is performed by querying the first lock object.

26. (Currently Amended) The computer readable storage medium of claim 15, wherein the method further comprises:

~~determining whether the data object was stored in the first lock object successfully, and upon unsuccessful storage, checking whether the data object assigned to the respective ID has been stored in the second storage location, and if the data object respective ID has not been stored, skipping deleting the data object from the first storage location and skipping deleting the ID from the first lock object.~~

27. (Currently Amended) The computer readable storage medium of claim 15, wherein the computer readable medium is provided as part of a computer program product.

28. (Currently Amended) A computerized system for processing data, the computerized system comprising:

a processor executing program instructions;

means for selecting a data object having an identifier (ID) from the first storage location;

means for storing the ID in a second lock object;

means for determining whether the ID is stored successfully, and upon a successful storage, storing the ID in a first lock object, thereby indicating that the data object is stored at the first storage location;

means for storing the data object, ~~the ID of which is contained in the first lock object,~~ at the second storage location;

means for deleting the data object, ~~the ID of which is contained in the first lock object~~, from the first storage location;

means for deleting the ID from the first lock object, thereby indicating that the data object is not stored at the first storage location, after the ~~respective data object assigned to that ID~~ has been deleted from the first storage location; and

means for deleting the ID from the second lock object after the ID has been stored in the first lock object.

29. (Previously Presented) The computer system of claim 28, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

30. (Currently Amended) The computer system of claim 28, wherein the data object is stored in a file and wherein an assignment of the ID to the file or to a name of the file, ~~in which the data object assigned to the ID is stored~~, is stored in the first lock object.

31. (Previously Presented) The computer system of claim 28, wherein the first lock object is stored on a nonvolatile storage means.

32. (Currently Amended) The computer system of claim 28, wherein the ID is stored in the second lock object after selecting [[a]] the data object having an ID from the first storage location ~~for the respective data object~~.

33. (Currently Amended) The computer system of claim 28, wherein the ID of the ~~selected data object~~ is stored in the second lock object before the data object assigned to that ID is stored at the second storage location.

34. (Currently Amended) The computer system of claim 28, wherein the means for storing the ID in a first lock object further comprises:

means for storing IDs of ~~all selected other~~ data objects in the first lock object before storing ~~any of the selected data object objects~~ at the second storage location.

35. (Currently Amended) The computer system of claim 28, further comprising:

means for checking whether the ID for the data object has been stored in the first lock object, and if the ID has been stored, skipping storing the data object at the second storage location.

36. (Previously Presented) The computer system of claim 28, further comprising:

means for checking whether the data object is contained in the second storage location, and if the data object is contained in the second storage location, skipping storing the data object at the second storage location.

37. (Previously Presented) The computer system of claim 36, wherein the means for checking comprises means for querying the first lock object.

38. (Currently Amended) The computer system of claim 28, further comprising:

~~means for determining whether the data object was stored in the first lock object successfully;~~

~~means for checking, upon unsuccessful storage in the first lock object, whether the data object assigned to the respective ID has been stored in the second storage location; and~~

~~means for skipping, if the data object respective ID has not been stored, deleting the data object from the first storage location and skipping deleting the ID from the first lock object.~~

39. (New) The method of claim 1, wherein storing the ID in the second lock object indicates that an action is being performed on the data object.

40. (New) The method of claim 39, wherein deleting the ID from the second lock object indicates that the action is not being performed on the data object.

41. (New) The computer system of claim 13, wherein storing the ID in the second lock object indicates that an action is being performed on the data object.

42. (New) The computer system of claim 41, wherein deleting the ID from the second lock object indicates that the action is not being performed on the data object.

43. (New) The computer readable storage medium of claim 15, wherein storing the ID in the second lock object indicates that an action is being performed on the data object.

44. (New) The computer readable storage medium of claim 43, wherein deleting the ID from the second lock object indicates that the action is not being performed on the data object.

45. (New) The computerized system of claim 28, wherein storing the ID in the second lock object indicates that an action is being performed on the data object.

46. (New) The computerized system of claim 45, wherein deleting the ID from the second lock object indicates that the action is not being performed on the data object.